Leicester Polytechnic

COMPUTER CENTRE

BBC FACTS SHEET

FS4 September 1984

CONTENTS

SECTION

		Page
1.	START UP	2
2.	LOADING, SAVING, AND RUNNING PROGRAMS FROM TAPE OR DISK	2
3.	FUNCTION OF SOME OF THE SPECIAL KEYS	4
4.	VARIABLES, NUMERIC ACCURACY AND RANGE	5
5.	ARITHMETIC OPERATORS	5
6.	INPUT FROM THE KEYBOARD	6
7.	PRINT FORMATTING	7
8.	SCREEN DISPLAY, MODES, AND MEMORY USAGE	8
9.	GRAPHICS AND COLOURS	9
1Ø.	VDU CODES AND CONTROL CHARACTERS	12
11.	USE OF PRINTER	13
12.	USE OF DISK DRIVE	13
13.	VOCABULARY	16
14.	ERROR MESSAGES	34

1. START UP

Check the cable linking the computer to the monitor (RGB port to input port), or to the television (UHF out to aerial socket). Check the power to the computer and monitor or tv screen. Check that the computer is switched on and that the monitor or tv is on.

The message:

```
BBC Computer 32K          or    BBC Computer 32K
BASIC                           Acorn DFS
>_                              BASIC
                                >_
```

should be displayed.

At this point the computer will be in mode 7 (teletext mode) ready for general use. The CAPS/LOCK key will be on because all commands to the computer must be in capital letters.

2. LOADING, SAVING, AND RUNNING PROGRAMS

On BBC Models A and B the default storage is tape. To use disk storage, type *DISK.

On the Model B Acorn DFS with default storage disk, type *TAPE to use tape storage.

The ESCAPE key allows you to leave cassette or disk operations at any point.

Loading a program from tape into the computer's memory

1) Insert the cassette into the tape recorder with the program side uppermost.

2) Check that the volume and tone controls are correctly set on the tape recorder, approximately 6 and 1Ø respectively.

3) Type LOAD"<filename>" and press RETURN, for example LOAD"WAGES".

4) The message 'Searching' will be displayed. Fast wind the tape either to just before the known start of the program on tape, or the beginning of the tape, press the PLAY button on the cassette recorder, and then press RETURN. When the computer finds the named program it will display 'Loading' and when the load is complete the > prompt will appear.

   LOAD"" instead of LOAD"<filename>" will load the first program that the computer finds on tape.

5) Type RUN and press RETURN to use the program. CHAIN"<filename>" or CHAIN"" will load and also run the program.

Loading a program from disk

1) Hold the disk at the label edge with the label facing upwards. For a single CUMANA drive unit, insert the disk into the drive unit and move the lever to the down position. For a double CUMANA drive unit, insert the disk into drive Ø, which will be positioned to either the left or the top of the pair of drive units, move the lever down, and set the switch to 4Ø or 8Ø depending on whether it is a 4Ø or 8Ø track formatted disk. All single drive units use a 4Ø track format, the double units can be either 4Ø or 8Ø.

2) Type LOAD"<filename>" and press RETURN, for example LOAD"WAGES". When loading a program from disk the filename must be specified.

3) When the load is complete the > prompt will appear.

4) Type RUN and press RETURN to use the program. CHAIN"<filename>" will load and also run the program.

5) To release the disk, turn the lever to the up position.

Some disks have a BOOT file on the disk. To use the BOOT file to load and run programs see Section 12 of this document.

Saving the current program on tape

1) Insert the cassette into the tape recorder.

2) Check that the volume and tone controls are correctly set on the tape recorder, approximately 6 and 1Ø respectively.

3) Type SAVE"<filename>", where <filename> is the chosen filename of up to 1Ø characters, and press RETURN.

4) The message "RECORD then RETURN" will be displayed. Rewind or forward wind tape to position for start of saving program.

5) Press the RECORD and PLAY buttons down on the cassette unit and then press RETURN on the BBC micro.

6) While the program is being copied to tape the name of the program and numbers will be displayed. When the save is complete the > prompt will appear.

7) *CAT can be used to verify that the program has been saved. The *CAT command will try to read the saved program back without destroying the program held in memory.

Saving the current program on disk

Refer to Section 12 for information on formatting disks for either single or double density use, drive and directory specification.

1) Insert a previously formatted disk into drive Ø of the drive unit and when it clicks into position close the drive door. The disk may already hold previously saved programs.

2) Type SAVE"<filename>" and press RETURN. The filename may consist of a maximum of 7 characters but must not include any of the following:

   # * . : or embedded spaces

   The red drive light will glow while the program is being saved.

3) Type *CAT to check that the file has been saved.

4) To release the disk, press lower half of the door inwards and the disk will pop out.

Running a program

When a program has been either typed in, or loaded into memory from tape or disk, the computer can be made to execute the program by typing the command RUN and pressing RETURN.

3. FUNCTION OF SOME OF THE SPECIAL KEYS

Where appropriate, the keys repeat if held down.

Key	Function
RETURN	Signals end of input line. Must follow every command, statement or data entry.
DELETE	Erase last character. Repeats if held down.
COPY	Used in conjunction with cursor control keys. Copies the character under the flashing cursor to the copy line.
CTRL	Used in conjunction with other keys to initiate special action.
ESCAPE	Stops a program running. Can be programmed to move from one part of a program to another.
BREAK	Soft resets computer but allows retrieval of existing program by typing OLD and pressing RETURN. Does not reset the clock, clear user-defined keys, or destroy the values of resident integer variables A% to Z%. Returns computer to mode 7.
TAB	Used under special circumstances such as word processing.
←↑↓→	Press any of these keys to enter EDIT mode. A block cursor marks the writing position on the screen. Move flashing cursor to editing position, press COPY key to move character under flashing cursor to line being written.
CAPS/LOCK	Gives capital letters and lowercase symbols, e.g. ABC ; :. Press again to release. Has own ON/OFF light.
SHIFT/LOCK	Gives capital letters and uppercase symbols, e.g. ABC + *. Press again to release. Has own ON/OFF light.
fØ to f9	User-definable function keys which will store one or several words.

The following keys generate ASCII characters in modes Ø to 6 and teletext display characters in mode 7.

Key symbol	Mode 7 teletext	Modes Ø to 6 ASCII
£	£	£
‾	‾	‾
^	↑	^ raise to a power
{	¼	{
}	¾	}
[	←	[delimit machine code section in program
]	→	]
\	½	\
¦	‖	¦ ¦ is the CTRL character ¦M inserts RETURN into user definition of red function keys.

4. VARIABLES, NUMERIC ACCURACY AND RANGE

All variable names can contain as many characters as required and all characters are used to identify the variable. A variable name must start with a letter and must not start with a BASIC keyword.

Integer variables are denoted by % at the end of the name, and string variables by $ at the end of the name. Arithmetic is performed faster with integer variables. The integer variables A% to Z% are special in that they are permanently allocated space in memory; they are called resident integer variables. If values are assigned to any of these variables they carry over from one program to another. The commands NEW, CLEAR, or RUN, or pressing the BREAK key, will not destroy values set in the resident integer variables, but switching off the computer will do so.

Type	Real	Integer	String
Name	NAME	NAME%	NAME$
Range from	$\pm$ 2x1Ø$^{-39}$	-2147483648	255 chars
to	$\pm$ 2x1Ø^{38}	+2147483648	
Accuracy	9 sig figs	1 digit	-
Stored in	4Ø bits	32 bits	ASCII values

Real, integer, and string array variables are also available; the naming convention is the same as the simple variables.

5. ARITHMETIC OPERATORS

Order of precedence is shown by the following group order, group 1 taking precedence over group 2 etc. The operators in any one group have equal priority and are dealt with on a left-to-right basis.

Group 1	Unary minus Unary plus NOT	Group 4	+ -
	Functions Brackets ( ) Indirection operators†	Group 5	= <> < > <= >=
Group 2	↑		
Group 3	* / DIV MOD	Group 6	AND
		Group 7	OR EOR

†Indirection operators are a more flexible variation of PEEK and POKE used for reading and writing to memory.

6. INPUT FROM THE KEYBOARD

INPUT

The INPUT command outputs a ? prompt and waits for string or numeric data as specified to be keyed in until the end of input is signalled by pressing the RETURN key. If numeric data is expected, a non-numeric character will return the number up to the invalid character only. A comma is used to separate data items so cannot be input as part of a string variable. Leading spaces are ignored.

Example	Purpose
10 INPUT A	Expects number from keyboard. ? prompt.
10 INPUT A$	Expects string from keyboard. ? prompt.
10 INPUT"CODE",A	Prints message 'CODE' followed by ?
10 INPUT"CODE"A	? prompt is suppressed if comma omitted.
10 INPUT A,A$	Expects number followed by string. Separate items input by comma or carriage return.

INPUT LINE

The INPUT LINE statement can be used in the same way as INPUT but it will accept everything that is typed including leading spaces and commas.

```
10 INPUT LINE"NAME",A$
```

GET and GET$

The GET and GET$ functions wait for a key to be pressed.

10 A$=GET$	Returns the character pressed in A$.
10 A=GET	Returns the ASCII value of the character pressed in A.

INKEY and INKEY$

INKEY and INKEY$ function as GET and GET$ except that they only wait a specified time for a key to be pressed. If a key is not pressed in that period then the program continues executing from the next line with value -1 for INKEY and "" for INKEY$. The unit of time is one hundredth of a second.

```
1Ø A$=INKEY$(5ØØ)          Will wait 5 seconds for a key to be pressed.
```

The command *FX 15,1 will flush the keyboard buffer.

7. PRINT FORMATTING

The first item is printed in a field 1Ø characters wide. Numeric variables are right justified in the field, string variables left justified. If the first item is numeric it can be forced to the left hand margin by a semi-colon before the numeric item. See examples 5 and 8 below.

A comma between items to be printed continues to print items in fields 1Ø characters wide. The number of fields will depend on the current mode, e.g. fields start at columns 1, 11, 21, 31 for mode 7 and columns 1, 11 for mode 5.

A semi-colon between items prints items packed close together.

A semi-colon at the end of the line makes the next print line continue on the same line.

@% allows the field width to be changed and the number of decimal places specified, for example @%=&2Ø3Ø9 specifies a field width of nine characters and three decimal places. The instruction can be broken up as follows.

```
&  - hexadecimal numbers follow
2  - format 2 for fixed decimal places (1 for exponential, Ø for default)
Ø3 - number of decimal places, maximum 9
Ø9 - field width
@% = 1Ø returns to default value, format Ø and field width 1Ø
     format Ø gives significant figures rather than decimal places
     format 1 gives figures as exponential values
```

Statement	Display (A$="POLY",A=5.6)
PRINT A$	POLY
PRINT A	5.6
PRINT A$,A,A$	POLY 5.6POLY
PRINT A,A$,A	5.6POLY 5.6
PRINT;A,A$,A	5.6 POLY 5.6
PRINTA$;A;A$	POLY5.6POLY
PRINTA;A$;A	5.6POLY5.6
PRINT;A;A$;A	5.6POLY5.6
MODE 7	
@%=&2Ø2Ø8:PRINT A,A	5.6Ø 5.6Ø
@%=&2Ø3Ø6:PRINT A,A	5.6ØØ 5.6ØØ

TAB(X) will move the cursor forward to column X on current line or next line down
TAB(X,Y) will move the cursor to character position X,Y on the screen

Superimposed characters

Characters can be superimposed on each other to generate special effects and characters, such as circumflex accents or true underlining. See Chapter 10 of the BBC manual.

Cursor control

Either VDU N or CHR$(N) can be used to move the cursor about the screen and clear the screen. See Section 10 for VDU codes.

The PRINT TAB(X,Y) to position the cursor is equivalent to VDU 31,X,Y.

Cursor OFF - VDU 23;8202;0;0;0
(VDU 23,1,0;0;0;0 version 1.0)
Cursor ON - MODE statement
(VDU 23,1,1;0;0;0 version 1.0)

8. SCREEN DISPLAY, MODES, AND MEMORY USAGE

Mode	Line/screen	Chars/screen line	No of colours	Resolution	Screen	Memory usage
0	32	80	2	very high	640x256	20K
1	32	40	4	high	320x256	20K
2	32	20	16	med	160x256	20K
3	25	80	text	-	-	16K
4	32	40	2	high	320x256	10K
5	32	20	4	med	160x256	10K
6	25	40	text	-	-	8K
7	25	40	teletext	-	-	1K

Note:

Model A has only modes 4 to 7 available.

Home position

Text - (0,0) at top left of text area
Graphics - (0,0) at bottom left of graphics area
horizontal, X-axis, 0-1279
vertical, Y-axis, 0-1023

Line length

Computer accepts up to 238 characters for one program line. To enter more than the width of a screen line, carry on typing at end of line without pressing the RETURN key.

9. GRAPHICS AND COLOURS

Monochrome monitors and television sets

If a monochrome monitor or television is used then a call to select a different colour will produce different shades of the default colour.

Window setting

A graphics window is set up with:

VDU 24,left X;bottom Y;right X;top Y;

where X,Y are graphics positions of the corners relative to the graphics origin.

A text window is set up with:

VDU 28,left X,bottom Y,right X,top Y

where X,Y are the character positions of the corners relative to the text origin.

Mode 7

Mode 7 uses standard teletext control codes to change colours, i.e. different codes and statements from those available in modes Ø to 6. Mode 7 allows limited but full colour graphics with economical use of memory. To change the colour of the text or background the control code must be sent via a PRINT CHR$(X) or VDU statement. The colour command is effective for that line only, each new line reverts to the default white on black background.

Code (X)	Colour of characters	Default for red function key
129	red	f1
13Ø	green	f2
131	yellow	f3
132	blue	f4
133	magenta	f5
134	cyan	f6
135	white	f7
136	flash on	f8
137	flash off	f9
14Ø	double height off	
141	double height character, must repeat program line	
145 - 151	colours as 129-135 for alternative lower-case graphic shapes	
157	new background	

Modes Ø to 6

The following keywords are used in producing high resolution graphics effects, MODE, GCOL, DRAW, MOVE, PLOT and VDU.

PLOT K,X,Y is a multi-purpose point, line, and triangle drawing statement. X,Y are the co-ordinates of the destination point and K is the manner of plotting.

K	
Ø	move relative to the origin
1	draw line from the origin to X,Y in current graphics foreground colour
2	as 1 but in logical inverse colour
3	as 1 but in current graphics background colour
4	move from last position to X,Y. Equivalent to MOVE.
5	draw line from last position to X,Y in current graphics foreground colour. Equivalent to DRAW.
6	as 5 but in logical inverse colour
7	as 5 but in current graphics background colour
8-15	as Ø-7 but with the last point in the line omitted in 'inverting actions'
16-23	as Ø-7 but with a dotted line
24-31	as 16-23 but without last point on line
32-63	reserved for Graphics Extension ROM
64-71	as Ø-7 but only a single point is plotted
72-79	reserved
8Ø-87	as Ø-7 but plot and fill triangle formed between last two points visited and point X,Y. PLOT 85,X,Y is the standard call.
88-255	reserved for future expansions

Position of origin may be changed using VDU 29.

To print a string at a specific screen position use TAB(X,Y) or join the graphics and text cursors with VDU 5.

Logical inverse colours

No of colours	Logical	Inverse
2	Ø	1
4	Ø	3
	1	2
	2	1
	3	Ø
16	Ø	15
	1	14
	:	:
	:	:
	:	:
	15	Ø

Colour selection

The normal colours for foreground and background for the different modes are shown in the table, they are known as the logical colours. The logical colour to be used is selected with the COLOUR X command, e.g. COLOUR 1. The sixteen possible colours are Ø to 15 of mode 2 and any logical colour may be changed by a VDU 19 statement to one of these actual colours.

VDU 19, logical colour, actual colour, Ø,Ø,Ø or ;Ø;
MODE 5: VDU 19,1,4,Ø,Ø,Ø or VDU 19,1,4;Ø; would change the default red foreground to actual colour 4, blue.

Modes	Foreground(X)	Background(X)	Colour
Ø,3,4,6	Ø	128	black
	1	129	white
1 and 5	Ø	128	black
	1	129	red
	2	130	yellow
	3	131	white
2	Ø	128	black
	1	129	red
	2	130	green
	3	131	yellow
	4	132	blue
	5	133	magenta
	6	134	cyan
	7	135	white
	8	136	flashing black-white
	9	137	flashing red-cyan
	10	138	flashing green-magenta
	11	139	flashing yellow-blue
	12	140	flashing blue-yellow
	13	141	flashing magenta-green
	14	142	flashing cyan-red
	15	143	flashing white-black

When selecting the colour for the graphics background or foreground, the method of placing the colour on the screen is also specified.

GCOL X,Y where Y is the logical colour, >127 defines the background colour, and X is Ø to 4, as below:

Ø plot the colour specified
1 OR the colour specified with that already there
2 AND the colour specified with that already there
3 Exclusive-OR the colour specified with that already there
4 invert the colour already there

10. VDU CODES AND CONTROL CHARACTERS

VDU statements are used to generate ASCII control codes which are interpreted by the VDU drivers to give easy access to text and graphics facilities from, for example, BASIC and assembler programs alike.

In direct mode the same effect can be produced by pressing the CTRL key and the specified key.

CTRL +	VDU Code	Effect
	Ø	does nothing
A	1	send next character to printer only
B	2	enable printer, send output to screen and printer
C	3	disable printer, output to screen only
	4	write at text cursor in normal fashion. MODE has same effect.
	5	graphics cursor only active. Text and graphics window in graphics colours. Screen does not scroll.
	6	re-enable VDU drivers after disabling with VDU 21
G	7	make short "beep" sound
H	8	cursor left one position
I	9	cursor right one position
J	1Ø	cursor down one position
K	11	cursor up one position
L	12	clear screen and home text cursor. CLS has same effect.
	13	carriage return to start of current line
N	14	page mode, e.g for long listings, SHIFT for next page
O	15	page mode off
P	16	clears the graphics area of the screen. CLG has same effect.
	17	define text colour. COLOUR has same effect.
	18	define graphics colour. GCOL has same effect.
	19	redefine logical colour
T	2Ø	restore default logical colours
U		delete whole of current line
	21	stop further output to screen, e.g hide password, VDU 6 re-enables
	22	select screen mode. MODE has same effect except VDU 22 does not reset HIMEM.
	23	reprogram display character. Not available for use in mode 7. VDU 23 can also be used to alter contents of 6845 CRTC circuit.
	24	define graphics window giving co-ordinates of four boundary points
	25	identical to PLOT. Use VDU 25 for machine code graphics.
Z	26	restore default windows and home cursor
	27	does nothing
	28	define text window
	29	define graphics origin
^	3Ø	home text cursor to top left of text area
	31	move text cursor to X,Y
	32-126	full set of letters and numbers in ASCII set
	127	backspace and delete, same effect as DELETE key
	128-223	undefined
	224-255	available for user-definition using VDU 23

11. USE OF PRINTER

1) Connect printer to computer

2) *FX 5,1 if using parallel printer (identified by flat ribbon of wires to printer)

   *FX 5,2 if using serial printer (identified by single flex to printer)

   default is parallel printer (i.e. no need to include *FX command)

   If serial printer selected, transmit baud rate to match that of the printer with a *FX 8,N command, where N is in range 1 to 8.

   1-75, 2-15Ø, 3-3ØØ, 4-12ØØ, 5-24ØØ, 6-48ØØ, 7-96ØØ, 8-192ØØ.

3) *FX 6,n is used to suppress the sending of particular characters, such as line feed, to the printer. The default is FX 6,1Ø to ignore line feeds, FX 6,Ø to reset to no suppression.

4) CTRL B (or VDU 2) to enable the printer, CTRL C (or VDU 3) to disable the printer.

Example set up for EPSON printer

*FX 5,2	select serial interface printer
*FX 8,4	select transmission rate of 12ØØ baud
*FX 6,Ø	for the EPSON - this may or may not be needed for your printer

Followed by VDU2 to enable the printer and, when finished, VDU3 to disable the printer.

Other useful commands might be VDU 1,15 for condensed characters and VDU 1,18 to cancel condensed characters. VDU 1,14 produces enlarged characters for one line only, but may be cancelled in mid-line with VDU 1,2Ø.

12. USE OF DISK DRIVE

All the disk filing system commands are prefixed with a * and are included in alphabetical order in Section 13.

Instructions for loading and running a program already on disk are given in Section 2. The commands are LOAD"<filename>" then RUN or CHAIN"<filename>" to load and run the program.

Formatting a new disk

A new disk requires formatting before it can be used to store programs or data. This process includes setting up the track and sector format on the disk and creating a catalogue.

The disk itself is the same for whichever disk unit you use.

This section will refer to formatting on the CUMANA drive units. The disk can be formatted with either 40 or 80 tracks. Each track has 10 sectors which can each hold 256 bytes of information. The single drive unit can only handle a 40 track format but the double drive unit can handle either the 40 track or more compact 80 track. In addition, the double drive unit can handle disks formatted on both sides.

In the dual drive unit the drives will be numbered 0 and 1 for single side format disks. Each side of a double side format disk will be given a drive number, 0 and 2 or 1 and 3.

To format a disk you will require both a new disk and also a disk carrying the format program, for example a UTILITIES disk. For a single drive unit, insert the UTILITIES disk in the drive and type:

*FORM40

TAKE THE UTILITIES DISK OUT OF THE DRIVE and insert your new disk into the drive. The progress of the formatting procedure is shown diagrammatically on the screen. When formatting is complete the option is given to format another new disk before leaving the format program.

For a double drive unit the UTILITIES disk must be inserted into drive 0 and the option is given to choose which drive you wish to format (0 to 3). Use the command *FORM80 for the 80 track format.

Catalogue

*CAT will list the catalogue of files on the current drive or the one specified, e.g. *CAT 1. The catalogue on a disk can be divided into independent directories. Files of the same name can be held on the same disk by putting each into a different directory. Each directory is identified by a single character, e.g. A or X. The default directory is $.

The catalogue listing shows the current directory file names first. The files on the other directories follow, each prefixed by its directory character, e.g. A.PHOTO. An L after a filename signifies that the file is locked. The catalogue heading shows the disk name, drive logged on to, drive and directory being accessed, and the option set for autostart.

At switch on, or after BREAK is pressed, the current drive defaults to 0 and the directory to $. Use *DRIVE and *DIR commands to log on to a different drive and/or directory.

When a program is saved it is automatically saved on the current drive and current directory unless the change(s) is shown in the filename specification, e.g. SAVE ":1.A.PHOTO".

Keep utility programs such as FORM40, sort programs etc together in one directory, referred to as the library directory. Utility programs are often in machine code and usually only require a *<filename> command to run them. See the BBC manuals for information on saving and running machine code programs.

File specification

The full file specification is:

:<drive>.<directory>.<filename>

For example:

CHAIN":Ø.L.WAGES"

If the drive is not specified then the current drive is assumed, Ø at switch on. If the directory is not specified then the current directory is assumed, $ at switch on. The filename can be made up of up to 7 characters but not including : . # or * or embedded spaces. If an embedded space is included, only the characters as far as the space are used. # and * are used when wanting to specify a group of files, the * being used to replace several letters or # to replace just one letter.

For example:

*INFO :Ø.#.WAGES will search all the directories on drive Ø and display information about any file called WAGES.

INFO :Ø.$.WA will search the $ directory on drive Ø and report on all files whose names begin WA.

INFO :1.$.#AG will search the $ directory on drive 1 for all filenames whose second and third letters are AG.

!BOOT

A disk may be set up with an autostart facility such that, when the SHIFT key is held down and the BREAK key pressed briefly, a particular program on the disk will automatically load and run. DO NOT RELEASE THE SHIFT KEY BEFORE THE BREAK KEY.

The instructions for the autostart are contained in a file called !BOOT which may be either a machine code or *EXEC file. For example, the WELCOME disk uses a !BOOT file to CHAIN the introductory program. A !BOOT file such as this can be created using the *BUILD command. The action initiated by the autostart is set with the OPT 4,n command at the time of creating the !BOOT file.

where	n=Ø	!BOOT is ignored (default)
	n=1	*LOAD !BOOT in memory
	n=2	*RUN !BOOT as a machine code program
	n=3	*EXEC !BOOT

The option set is shown in the catalogue heading of that disk, by default it is set to Ø to ignore !BOOT.

Copying, renaming, and removing programs

*COPY <source drv><dest drv><filename> will copy a program from one drive to another.
*COPY Ø 1 WAGES will copy the program WAGES on drive Ø to drive 1
*COPY Ø 1 * will copy all programs on drive Ø to drive 1
CAUTION the COPY programs may overwrite the contents of memory so save your current program before using *COPY.

*BACKUP<source drv><dest drv> is a sector by sector copy program for dual or single drive disk units.
*BACKUP must be preceded by *ENABLE
*BACKUP Ø Ø will create a backup on a single drive unit, instructions are given as it runs
*BACKUP 1 Ø will create a duplicate on drive Ø of the disk in drive 1.
CAUTION as for *COPY

*RENAME <old filename><new filename> will change the filename and, if requested, move it to another directory on the same disk.
*RENAME WAGES B.PAYSLIP will rename WAGES as PAYSLIP and move it to directory B

*DELETE <filename> will remove a single named file from the catalogue of a disk.
*DESTROY <matchname> is used to remove a group of files in one operation from a disk. The command must be preceded by *ENABLE.
*WIPE is used to remove a group of files one at a time, confirmation is sought for removing each one.
*DELETE, *DESTROY and *WIPE will not remove locked files. *ACCESS is used to lock or unlock a file.

13. VOCABULARY

All keywords must be in capital letters.

If a BASIC word starts with a * then the whole of the rest of the statement is passed to the operating system directly.

Many of the commands have permissible abbreviations which are listed in the manual. The abbreviations which are likely to be of most use are:

*. for *CAT
L. for LIST
P. for PRINT

[ ] enclose optional items

KEYWORD	EXAMPLE	PURPOSE
@%	@%=&2Ø2Ø9	Change default width of print field and specify number of decimal places. See Section 7.
ABS	1Ø A=ABS(B)	Absolute value of B (ignores sign).

KEYWORD	EXAMPLE	PURPOSE
*ACCESS		Locks or unlocks a named file to prevent it being deleted or overwritten.
	*ACCESS WAGES L	Locks the file WAGES.
	*ACCESS WAGES	Unlocks the file WAGES.
ACS	1Ø X=ACS(Y) 1Ø ANGLE=DEG(ACS(Ø.5678))	Arc-cosine, calculates the angle (in radians) whose cosine is known. Convert radians to degrees with function DEG.
ADVAL	67Ø X=ADVAL(3)	Analogue to digital converter value.
	995 TEMP=ADVAL(X)	Returns the last known value of the analogue to digital channel specified in brackets. Channels 1 to 4.
	X=ADVAL(Ø)	Special function to test which FIRE button pressed on games paddle.
	X=ADVAL(-1)	Negative argument in range -1 to -7 tests various internal buffers in computer.
AND	IF X=5 AND Y>Ø THEN---	Use as logical operator, both conditions must be met or bit-wise comparison of two numbers.
ASC	1Ø A=ASC(B$)	Returns integer value corresponding to the ASCII code of the first character in B$. A null string returns the value -1.
ASN	9Ø ANGLE=ASN(Y) 9Ø X=DEG(ASN(Ø.3456))	Arc-sine. Calculate the angle in radians whose sine is known.
ATN	9Ø X=ATN(2.31)	Arc-tangent. Calculate the angle in radians whose tangent is known.
AUTO		Automatically generates program line numbers.
	AUTO	Start at line 1Ø, ascend in steps of 1Ø. Highest line number 32767. Press ESCAPE to leave AUTO.
	AUTO,15Ø	Start at line 15Ø, steps of 1Ø.
	AUTO,5Ø,5	Start at line 5Ø, step 5.

KEYWORD	EXAMPLE	PURPOSE
*BACKUP	*ENABLE *BACKUP<sdr> <ddr>	Sector by sector copy program. The *ENABLE command must precede the *BACKUP.
	*BACKUP Ø 1	To copy the disk in drive Ø on to the disk in drive 1.
	*BACKUP Ø Ø	To make a backup copy using a single drive unit. Instructions given as backup proceeds.
BGET#	11Ø Y=OPENIN"RESULTS" 12Ø CHAR=BGET#(Y)	Reads a single (byte) from a file on cassette tape or floppy disk on channel allocated. The file must have been previously opened with OPENIN command.
BPUT#	12Ø X=OPENOUT"RESULTS" 33Ø BPUT#Y,32	Writes a number (byte) to a file on cassette tape or floppy disk on channel allocated. File must have been previously opened. The number must be in the range Ø to 255. Numbers larger than 256 will be brought to within required range by repeated subtraction of 256. For large numbers use PRINT#.
*BUILD	*BUILD !BOOT	Creates a file directly from the keyboard. After typing the command, everything else entered will go into the named file, !BOOT. Line numbers automatically displayed on screen. Use ESCAPE key to terminate *BUILD.
CALL	5Ø CALL small,J,K,L	Transfer control to a machine code subroutine. Parameters, e.g. J, K, L, can be passed.
*CAT	*CAT	Displays catalogue of files on cassette/disk/net. Can be used to verify a program just saved on tape, see Section 2.
	*CAT 1	Displays catalogue of files on disk in drive 1. See Section 12 for more information.

KEYWORD	EXAMPLE	PURPOSE
CHAIN		Enables one program to load and run another program automatically. Clears all variables except @% and A% to Z%.
	15Ø CHAIN"UNIT2"	Will load and run program called "UNIT2" from tape or disk.
	5ØØ CHAIN""	Will load and run the next program on cassette tape.
CHR$	1Ø A$=CHR$(34)	Returns as a string the ASCII character with the numeric value specified, in this case ".
	11Ø PRINT CHR$(13)	Prints the character with the ASCII code specified, in this case a carriage return. (See VDU also.)
CLEAR		Resets all the variables to Ø or null except for the resident integer variables A% to Z%.
CLOSE#	99Ø CLOSE#5	Close file 5.
	99Ø CLOSE#Ø	Will close all files.
CLG	55Ø CLG	Clears the graphics screen leaving background colour as selected by GCOL. Move cursor to home position (Ø,Ø) at bottom left of graphics area.
CLS	56Ø CLS	Clears the current text area leaving background colour as selected by COLOUR. Move cursor to home position (Ø,Ø) at top left of current text area.
COLOUR	MODE 5:COLOUR 1	To change the colour of the text to selected number.
	COLOUR 129	To change the colour of the background. See separate table for colour possibilities.
*COMPACT	*COMPACT 1	Moves all available space on a disk into one continuous block following the current files. Particularly useful if disk nearly full. WARNING This command may overwrite the contents of memory; save program or data in memory before issuing the command.

KEYWORD	EXAMPLE	PURPOSE
*COPY	*COPY Ø 1 WAGES	Copy the file WAGES from drive Ø to drive 1. WARNING This command may overwrite the contents of memory.
COS	1Ø A=COS(B) 11Ø A=COS(RAD(45))	Cosine of B. B must be in radians.
COUNT	11Ø A=COUNT PRINT COUNT	Returns the number of characters printed to screen, printer etc. since the last new line.
DATA	DATA 2,FRED,ERIC DATA " JONES","SMITH,D."	Specifies data to be accepted by a READ statement. Strings only need to be delimited by " if they contain leading spaces or commas. If data is text where numeric data was expected, then Ø returned.
DEF	9Ø9Ø DEF FNVAT(g)=1.15*g 23Ø PRINT FNVAT(P)	Defines function or procedure which can then be called by name. Multiple line definitions possible, last line of function starts with =, last line of procedure is ENDPROC.
DEG	1ØØ X=DEG(1.36)	Converts angles expressed in radians to degrees.
DELETE		Direct command to delete the set of lines specified.
	DELETE Ø,6Ø	Deletes lines Ø to 6Ø inclusive.
	DELETE 5ØØØ,32767	Will delete line 5ØØØ to end as 32767 highest possible line number.
*DELETE	*DELETE FRED	Removes file FRED from current directory on current drive.
*DESTROY		Used to destroy a group of files on a disk.
	*ENABLE	*ENABLE required before *DESTROY.
	*DESTROY *.H*	Will destroy any files with filenames containing the substring .H, e.g. A.HELLO

KEYWORD	EXAMPLE	PURPOSE
DIM	DIM CODE(100),PA$(20) DIM NAME$(4,15) DIM X 24	Specifies maximum size of arrays, single or multidimensional. First element in array is numbered 0, no maximum within bounds of memory. DIM statement initialises numeric array variables to 0 and strings to null strings. DIM can also be used to reserve bytes in memory for special applications. In example reserves 25 bytes at address X.
*DIR	*DIRA	To change the current directory to <dir>, in this example A. The current disk directory is always set to '$' when BREAK is pressed.
*DISC *DISK	*DISC *DISK	Select disk file system for file operations such as LOAD and SAVE.
DIV	120 XX=20DIV 3	Returns the whole number part of the result of a division.
DRAW	560 DRAW X,Y	In modes 0, 1, 2, 4 draws a line from current point to point given by co-ordinates X,Y in current foreground colour.
*DRIVE	*DRIVE 1 *DRIVE 0	Sets the current drive to drive 1. Sets the current drive to drive 0.
*DUMP	*DUMP TESTA	Produces a hexadecimal listing of the named file on the screen.
*ENABLE	*ENABLE	Must be given immediately before the *BACKUP or *DESTROY commands to prevent them being used accidentally.
END	1500 END	Optional end-of-program statement, use wherever and whenever required.
ENDPROC	1000 DEF PROCFIG(K) : 1090 ENDPROC	Final statement of PROCEDURE definition.

KEYWORD	EXAMPLE	PURPOSE
ENVELOPE		Used with the SOUND statement to control the volume and pitch of a sound. 14 parameters required, see manual.
EOF#	100 X=EOF#(N) 200 REPEAT UNTIL EOF#(N)	Returns -1 if end of file reached otherwise returns 0. N is the channel number.
EOR	200 R=X EOR Y	Performs the operation of logical bitwise exclusive-or between the two operands.
ERL	8500 X=ERL	Returns the line number of the line where the last error occurred.
ERR	1000 WRONG=ERR 100 IF ERR=17 THEN--- PRINT "YOU CANT ESCAPE!"	Returns the error number of the last error which occurred.
EVAL	10 A$="X 2+X-3" 20 FOR X=1 TO 5 30 Y=EVAL(A$):PRINT Y 40 NEXT X.	Allows the user to input, e.g. a mathematical expression into a string (A$). EVAL passes the function into the program.
*EXEC	*EXEC"SHORT"	Used to merge the program "SHORT", previously saved by *SPOOL, into the currently held program.
EXP	10 A=EXP(B)	Returns value of e^B.
EXT#	100 X=EXT#(employ)	Returns the length in bytes of the file opened on channel given. For use with disc and network file system only.
FALSE	PRINT 5=4 245 UNTIL FALSE 20 CLOCKSET=FALSE	Returns the value 0 if statement is not true. Numerical value of FALSE is 0.
FN	1000 DEF FNVAT(g)=1.15*g	Reserved word used at start of all user-defined functions.
FOR--NEXT	10 FOR X%=1 TO 5 : 50 NEXT X% :	Perform the block of instructions between the FOR and NEXT statements. Step is default of 1 unless specified.
FOR--STEP--NEXT	70 FOR X=5 TO 1 STEP-.5 : 90 NEXT X.	Integer control variables are faster than real control variables.

KEYWORD	EXAMPLE	PURPOSE
*FX	*FX9,10 *FX21,0 *FX8,4	Used to control a large number of the computer's special effects such as flashing colours or flushing buffers. See manual for details and Section 11 for use of printer.
GCOL	100 GCOL 0,2 GCOL 3,129	Select colour to be used by subsequent graphics operations. First number specifies mode of action, second the colour. See Section 9.
GET	1040 keybit=GET 350 X=GET	Waits for a key to be pressed and returns ASCII number of key pressed. Does not display character key pressed.
GET$	1050 A$=GET$ 2010 IF GET$="Y"THEN---	Waits for a key to be pressed then returns string containing the character. Does not display character key pressed.
GOSUB	110 GOSUB 350	Execute subroutine which starts at line 350. Maximum nested depth is 26.
GOTO	GOTO 100 110 IF---THEN GOTO 300 110 GOTO (X*5+2)	Start program at line 100 without destroying values assigned to the variables. Transfer control to a specified or calculated line number.
*HELP	*HELP DFS *HELP UTILS	Will list the disk filing system commands. Will list the disk utilities available.
HIMEM	100 HIMEM=HIMEM-40 100 PRINT HIMEM	HIMEM contains the address of the first byte that BASIC does not use.
IF--THEN-- IF--THEN--ELSE--	100 IF Y THEN A=B 150 IF X$="Y"THEN 450 150 IF X$="Y"THEN--- 160 ELSE---	If test condition true then do whatever follows. If test condition not true then either perform the ELSE statement if present or drop through to the next line.

KEYWORD	EXAMPLE	PURPOSE
*INFO	*INFO A.HELLO	Display detailed file information in following order: directory, filename, access, load address, execution address length in bytes, start sector.
INKEY	1ØØ NUM=INKEY(X)	1) Waits a specified time (X) for a key to be pressed, then returns ASCII value of key, if no key pressed returns -1. X measured in hundredths of a second. *FX 15,1 to flush the buffer.
	IF INKEY(-87)THEN---	2) INKEY with negative number checks for a particular key, example is testing for key L.
INKEY$	12Ø LT$=INKEY$(X) 12Ø RST$=INKEY$(1ØØ)	Waits a specified time for a key to be pressed then returns key character pressed in a string variable. If no key pressed returns null string. X measured in hundredths of a second. *FX 15,1 to flush the buffer.
INPUT		Used to request input from the user.
INPUT LINE		See Section 6 for details.
INPUT#	INPUT#X,A,B$	Used to read data back into the computer from cassette or disk. X is channel number.
INSTR	1Ø X=INSTR(A$,B$,Z) 1Ø X=INSTR("HOUSE","OU") 1Ø X=INSTR(A$,"OU",3)	Returns the start position of a substring (B$) within a string (A$). Ø is returned for no match. 1 is returned if searching for null string. Starting position (Z) of search may be specified.
INT	15Ø X=INT(3.1) 15Ø X=INT(-3.1)	Converts a real number to the lower integer. Returns 3. Returns -4.
*KEY	*KEY 2 DATA	Programs one of the user-defined keys. Will program key f2 to be "DATA".

KEYWORD	EXAMPLE	PURPOSE
LEFT$	1Ø A$=LEFT$(B$,C)	Returns the C leftmost characters of the string B$.
	1Ø A$=LEFT$("HOUSE",2)	Would return "HO".
LEN	1Ø A=LEN(B$)	Returns the number of characters in a string.
LET	1Ø X=X+6 or 1Ø LET X=X+6	The word LET is optional when assigning a value to a variable.
*LIB		Sets the drive/directory containing the library of (utility) programs.
	*LIB :1.A	Sets the library to drive 1 directory A.
LIST		Direct command only, cannot be used in a program.
	LIST	Lists current program.
	LIST 115	Lists line 115.
	LIST ,3ØØ	Lists lines Ø to 3ØØ.
	LIST 3ØØ,	Lists 3ØØ to end.
*LIST	*LIST MYTEXT	Displays the textfile MYTEXT on the screen with line numbers.
LISTO		LISTO affects the print format produced by subsequent LIST commands. Allows insertion of spaces: a) after line number b) during FOR--NEXT loops c) during REPEAT--UNTIL loops
	LISTOØ	Ø - no inserted spaces
	LISTO1	1 - insert space after line number
	LISTO2	2 - FOR--NEXT loops
	LISTO4	4 - REPEAT--UNTIL loops
	LISTO6	Ø, 1, 2, 4 can be added to produce number in range Ø to 7 for combined options. LISTOØ option advised when editing a file.
LN	1ØØ X=LN(temp)	Returns natural logarithm of argument.

KEYWORD	EXAMPLE	PURPOSE
LOAD	LOAD"STATS2"	Direct command to load the named program from tape, disk or network. Deletes current program, and clears all variables except A% to Z%.
	LOAD""	Will load the next program from tape only. See CHAIN for loading and running a program from within another program.
	LOAD ":1.A.WAGES"	Load WAGES from disk drive 1, directory A.
*LOAD	*LOAD"PROG"7E8Ø	Load a machine code program. If an absolute load address is given it will force the program to load at that address.
LOCAL	9ØØ DEF PROCtriang (sz) 9Ø5 LOCAL X1,X2,Y1,Y2 : 955 ENDPROC	LOCAL can only be used inside a function/procedure definition. LOCAL saves the values of the external variable names and restores these original values when function/procedure completed.
LOG	1ØØ X=LOG(Y)	Logarithm (base 1Ø) of Y. Anti-logarithms obtained by using Y=1Ø↑X.
LOMEM	PRINT LOMEM LOMEM=TOP+&1ØØ	Start address for storage of variables other than A% to Z% and @%. Normally set to be the same as TOP.
MID$	1Ø A$=MID$(B$,C,D)	Returns D characters from B$ starting at C.
	1Ø A$=MID$("HOUSE",2,3)	A$="OUS".
	1Ø A$=MID$("HOUSE",2)	A$="OUSE".
MOD		Returns the remainder after division of integer numbers, decimal numbers are truncated to integers.
	PRINT 2Ø MOD 3	Would return 2.
	PRINT 1Ø.1 MOD 3.8	Would return 1.
MODE	1Ø MODE 5	Clears the screen and sets up the mode specified. See separate section for details. Modes 4 to 7 only available on Model A computer. MODE cannot be changed inside a procedure or function. MODE resets the value of HIMEM.

KEYWORD	EXAMPLE	PURPOSE
*MOTOR	*MOTOR Ø *MOTOR 1	Used to turn the cassette motor relay on and off. Ø for off.
MOVE	2ØØ MOVE 15Ø,5ØØ MOVE X,Y	Moves the graphics cursor to a new position without drawing a line.
*NET	*NET	Selects network file system for future file operations.
NEW	NEW	Command to reset internal pointers and variables, except A% to Z% and @%, to a "no program present" condition. See OLD to recover the program.
NEXT	1ØØ FOR J=1 TO 1Ø : 15Ø NEXT	The closing statement of a FOR--NEXT loop.
NOT	1ØØ IF NOT(A=2Ø AND B=3) THEN 35Ø	Reverses the condition of a test. See separate section for priority.
OLD	OLD	Will retrieve a program after a NEW command or pressing the BREAK key provided no program lines have been entered or new variables created. Check first line number correct if > 255.
ON--GOTO-- ON--GOSUB-- ON ERROR GOTO ON ERROR GOSUB	1ØØ ON A GOTO 15Ø, 25Ø,35Ø 1ØØ ON A GOSUB 1ØØØ, 2ØØØ,3ØØØ 5ØØ ON ERROR GOTO 7ØØØ 9ØØ ON ERROR OFF	Controls next line or subroutine to be executed depending on value of A. Value of A must range from 1 upwards in steps of 1. If A outside range then execution drops through to next line. ELSE can be used to trap out-of-range values. Cancels the last ON ERROR GOTO.
OPENIN	5Ø X=OPENIN("TABLES")	Opens a channel (X) ready for use to read data into the computer from the named file held on disk or cassette. While the file remains open the channel number is constant.
OPENOUT	7ØØ X=OPENOUT("TABLES")	Opens a new file on cassette or disk ready to receive data from the computer. If a file of the same name exists it will be deleted then a new file of that name created.

KEYWORD	EXAMPLE	PURPOSE
OPT	200 OPT 1 350 OPT(pass*2+list)	Determines what output is produced on the screen when assembly language routines are processed by the BASIC interpreter.
*OPT	*OPT 1, X	Enables a message system to display file information each time a disk file is accessed. 1<X<99.
	*OPT 1, 0	Displays the message system.
	*OPT 4, X	Changes the start-up option of a disk 0<X<3. 0 does nothing, 1 will LOAD !BOOT, 2 will *RUN !BOOT, 3 will *EXEC !BOOT.
OR	10 IF A=20 OR B=3 THEN 100	Either condition may be met.
PAGE	10 PAGE=&5000 20 PRINT ~PAGE PRINT PAGE	PAGE gives start address of user's program in memory. Can be used to allow simultaneous storage of two programs in different areas of memory. Use with care.
PI	10 A=PI*P↑2 PRINT PI	The constant pi, 3.14159265.
PLOT	100 PLOT K,X,Y 200 PLOT 6,100,200	The multi-purpose point, line and triangle drawing statement in BASIC. K gives the manner of drawing to the point X,Y. See other section for possible range and effect of K. MOVE is equivalent to PLOT 4, DRAW is equivalent to PLOT 5.
POINT	1340 colour=POINT(X,Y)	Returns the logical colour of the screen at the graphic point specified. If the point is off the screen then the function returns -1.
POS	750 X=POS	Returns the horizontal position of the cursor in the current text window. Range is 0 to 19, 39 or 79 depending on MODE selected.
PRINT		The PRINT statement governs the display of material on the screen. See Section 7 for details.

KEYWORD	EXAMPLE	PURPOSE
PRINT#	PRINT# X,A,B,C$	Statement used to write data to a file on tape or disk. The file must have been previously opened using an OPENIN or OPENOUT statement. The channel number, X, must match that in the OPENIN statement.
PROC	10 DEF PROCdraw 100 PROCyear	A reserved word used to prefix a procedure name.
PTR#	PRINT PTR#X 560 PTR#FL=PTR#FL+80	Pointer used to select an item to be read from a serial file. PTR# cannot be used with tape systems.
RAD	PRINT RAD(45)	Converts an angle measured in degrees to radians.
READ	10 READ A,B$: : 990 62,"VAT"	Copies the next item from a data-list into the variable or variables which follow the keyword READ. The data type must match the variable type.
REM	REM NOW CALC. AREA	Allows comments to be inserted into a program. The computer ignores any statement which begins REM.
*RENAME		Changes the file name and moves it to another directory if required.
	*RENAME SUMS B.ADD	The file $.SUMS becomes B.ADD
RENUMBER		Direct command to renumber the lines of a program. GOTO etc destinations will be amended accordingly. GOTO a missing line number will produce error message. It cannot handle calculated line numbers in ON-GOTO.
	RENUMBER	Will renumber from 10 in steps of 10.
	RENUMBER 500	Will renumber from 500 in steps of 10.
	RENUMBER 100,5	Will renumber from 100 in steps of 5.

KEYWORD	EXAMPLE	PURPOSE
REPEAT--- UNTIL	10 REM PRINT STARS 20 NOW=TIME 30 REPEAT PRINT"*"; 40 UNTIL TIME=NOW+100	Will repeat that program section until the given condition is met. Always executes once and may be nested up to a depth of 20. A single REPEAT may have more than one UNTIL.
REPORT	500 REPORT	Prints the error message associated with the last error condition.
RESTORE	500 RESTORE 500 RESTORE 3500	Moves the data pointer back to the first DATA statement in the program or to the DATA statement at the line given.
RETURN	200 RETURN 500 IF X$="N"THEN RETURN	The word RETURN is used to signal the end of a subroutine section and returns the program flow to the statement after the GOSUB call.
RIGHT$	10 A$=RIGHT$(B$,C) 10 A$=RIGHT$("HOUSE",2)	Returns the C rightmost characters of the string B$. Would return "SE".
RND	RND RND(1) RND(0) RND(X) RND(-X)	Generates a random number in the range -2147483648 to 2147483647. Generates a number between 0 and 0.999999. Repeats the last random number given by RND(1). Generates a number between 1 and X. Returns -X and resets the generator to a number based on X.
*ROM	*ROM	Selects ROM cartridge file system.
RUN	RUN	Use in direct mode or program mode to make the computer execute the current program. All variables are reset except @% and A% to Z%.
*RUN	*RUN"match"	Loads and executes a machine code program. Current directory searched first then library directory.

KEYWORD	EXAMPLE	PURPOSE
SAVE	SAVE"STATS/V1" SAVE A$	Saves current program with filename "STATS/V1" on cassette unit. The filename must start with a letter, maximum 1Ø characters, no embedded spaces or punctuation marks.
*SAVE	*SAVE"patch"6ØØØ 62ØØ *SAVE"match"4CØØ 4CE9	Save a section of machine code program at start address (SSSS), end address +1, (FFFF or length +LLLL). An execution address EEEE may be given otherwise taken as start address.
SGN	A=SGN(B)	Returns -1 if B is negative, Ø if B is zero, or +1 if B is positive.
SIN	1Ø A=SIN(B) 11Ø value=SIN(RAD(45))	Returns sine of B, B in radians. Use RAD to convert degrees to radians.
SOUND	SOUND A,B,C,D SOUND 1,-15,53,2Ø	Generates sound through internal speaker in accordance with parameters. A is sound channel, B is envelope number, C is pitch, and D is duration of note.
SPC	PRINT A$;SPC(6);A INPUT SPC(1Ø);"VALUE",V	Prints number of spaces specified in brackets to the screen. Use only with PRINT and INPUT.
*SPOOL	*SPOOL "VERS1"	All material subsequently displayed on the screen, e.g. a program listing will also be written to the named file.
SQR	A=SQR(B)	Returns the square root of B.
STEP		Part of FOR--TO--STEP---NEXT structure. STEP is optional, default value is STEP=1.
STOP	99Ø IF X$="**"THEN STOP	Interrupts a program run and prints message STOP at line XXXX.
STR$	1ØØ A$=STR$(B)	Converts a number into the equivalent string representation. STR$ is affected by the field width and format constraints set in @%.

KEYWORD	EXAMPLE	PURPOSE
STRING$	10 A$=STRING$(40," ") 90 B$=STRING$(10,"*0")	Produces multiple copies of a shorter string.
TAB	90 TAB(X)	Move cursor forward to column X. On current line or next line down.
	90 TAB(X,Y)	Move cursor to character position X,Y on screen.
TAN	10 A=TAN(B) PRINT TAN(RAD(45))	Returns tangent of B, B in radians. Use RAD to convert degrees to radians.
*TAPE	*TAPE	Selects the cassette filing system running at 1200 baud for future file operations.
	*TAPE3	Select cassette filing system at 300 baud.
TIME	TIME=((HO*60+mi)*60 +Se)*100 410 nowtime=TIME	Used to set or read the internal timer which counts in 1/100 second intervals.
*TITLE	*TITLE "DISK1"	Changes the title of disk in current drive to "DISK1 ". Title is padded with spaces to 12 character length.
TOP	200 PRINT TOP 540 X=TOP	Returns the first free memory location after the user's program.
TRACE	TRACE ON	Causes the interpreter to display executed line numbers when it encounters them.
	TRACE 1000	Only line numbers less than 1000 will be displayed.
	TRACE OFF	Turns trace mode off.
TRUE	300 UNTIL result=TRUE PRINT 4=4	TRUE is represented by the value -1 in the computer.
*TV	*TV	Controls the vertical position of the screen display and the picture interlace.
*TYPE	*TYPE MYPROG	Lists the named file on the screen without line numbers.
UNTIL		See REPEAT--UNTIL.

KEYWORD	EXAMPLE	PURPOSE
USR	14ØØ R=USR(&3ØØØ)	Function allowing machine code to directly return a value for problems which do not require the flexibility of CALL.
	67Ø result=USR(plot 5)	
VAL	A=VAL(B$)	Returns numeric value of string B$. If string not numeric then returns Ø.
VDU		Used to generate a sequence of numbers that are then sent to the VDU drivers. Equivalent to PRINT CHR$ except that it does change value of COUNT.
	VDU 28,Ø,5,39,Ø	Would define a six line text window at top of screen - mode 5
	VDU 24,Ø;Ø;1279;83Ø;	Would define a graphics area at bottom of screen and 83Ø points high.
	VDU 2	Turn printer on. See Section 1Ø for VDU codes.
VPOS	15Ø vert=VPOS	Returns the vertical position of the text cursor.
WIDTH		Used to set the overall "page width", initially set to zero which the interpreter interprets as unlimited. Affects all output to the screen or printer.
	1ØØ WIDTH 35	Forces a new line after 35 characters printed.
*WIPE		Removes specified files from the catalogue and rearranges the catalogue. Asks for confirmation before removing a file.
	*WIPE *.SU*	Deletes all files on current drive beginning with the letters SU.

14. ERROR MESSAGES

When an error occurs the computer will print out an error message and set the two variables ERR (error number) and ERL (error line number). The message can be suppressed by using

ON ERROR GOTO 9000

and providing special action routines starting at line 9000. Some of the most common error messages are listed below but the full list is given in the BBC manuals.

No	Message	Comments
10	Bad DIM	An array must hold a positive number of elements.
	Bad program	This is an untrappable error and is usually caused by a read error or by only loading part of a program or by overwriting part of the program in some way.
18	Division by zero	Check any variables used as the division never have the value 0. Note that when a variable is declared as LOCAL it is set to 0.
	Failed at <line number>	Occurs when a program is renumbered and the original contains a destination line number which does not exist.
4	Mistake	The computer could not make any sense of the input line.
0	No room	All available memory used up.
26	No such variable	All variables must be assigned a value, or made LOCAL, before they can be used. Will also occur if illegal spaces included, for example TAB (10) instead of TAB(10) will cause the computer to consider TAB a variable name.
15	Subscript	Occurs if the program tries to access an array element less then zero or greater than the range set in the DIM statement.
16	Syntax error	A command was terminated incorrectly.

Common disk error messages

Message	Comments
Bad filename	Invalid filename, maximum length is 7 characters.
Disc full	Insufficient space on the disc to open or save the specified file. DO NOT USE *COMPACT until the program in memory has been saved in some other manner as *COMPACT can corrupt the current program in memory.
Disc fault NN at TT SS	The computer cannot read the disk. The disk might be unformatted, damaged, write protected or of wrong type, or the disk drive might be faulty.
Not enabled	The command *ENABLE must precede a *BACKUP or *DESTROY command
File open	File already open or trying to delete an open file. If a program under test crashes leaving one or more files open then a CLOSE Ø will close any files still open.
Disk fault 18 at XX/XX	Cumana double drive unit, switched to 8Ø track for 4Ø track formatted disk.